

# **Työaihekanta tietokantasovelluksen dokumentaatio**

Andreas Andra

Helsinki 30.8.2017

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

# Sisältö

<b>Johdanto</b>	<b>3</b>
<b>Yleiskuva järjestelmästä</b>	<b>4</b>
<b>Järjestelmän tietosisältö</b>	<b>7</b>
<b>Relaatiotietokantakaavio</b>	<b>10</b>
<b>Järjestelmän yleisrakenne</b>	<b>11</b>
<b>Käyttöliittymä ja järjestelmän komponentit</b>	<b>12</b>
<b>Asennustiedot ja käyttöohjeet</b>	<b>13</b>
<b>Testaus, tunnetut bugit ja puutteet, jatkokehitysideat &amp; muu dokumentaatio</b>	<b>14</b>

## 1. Johdanto

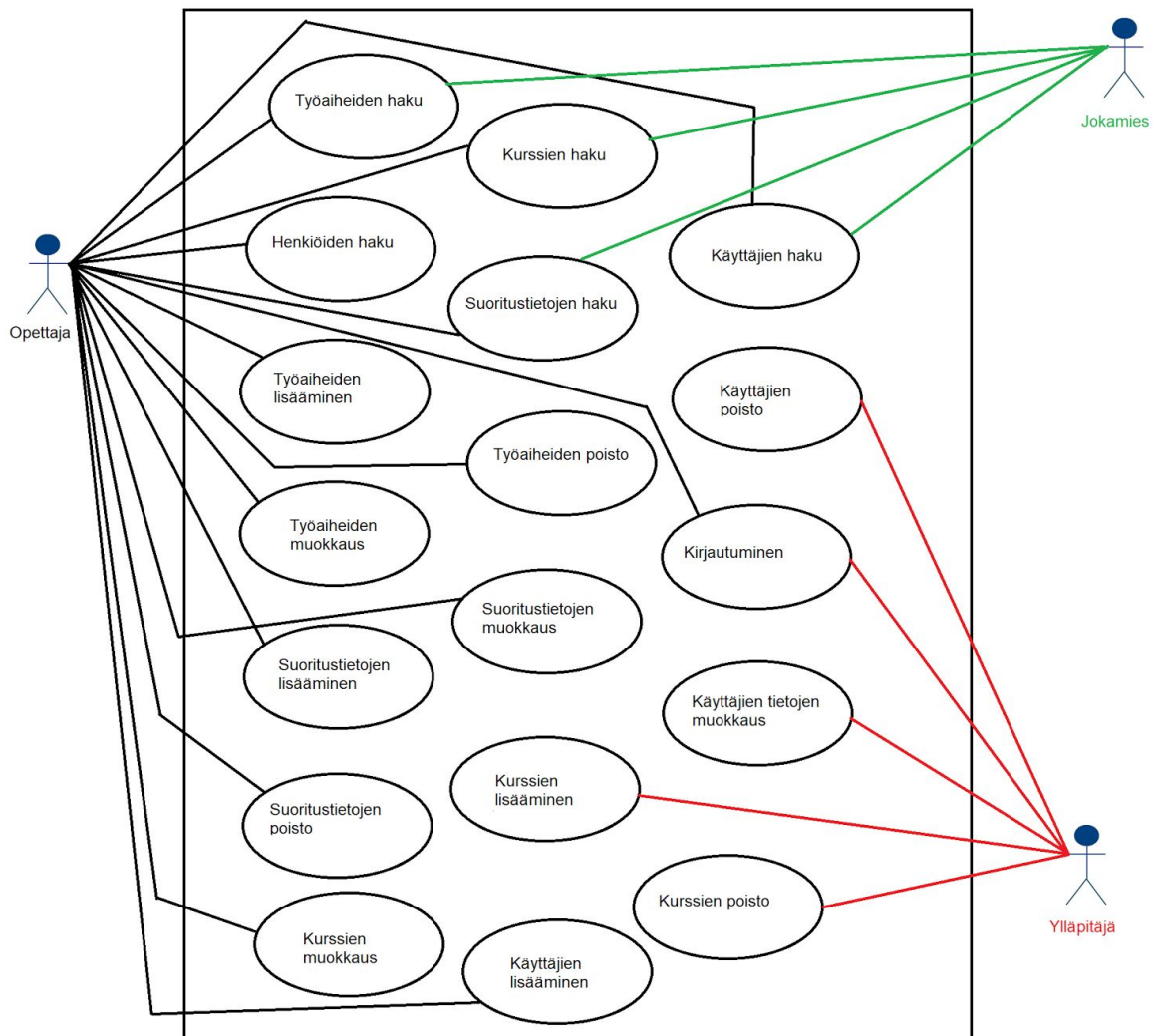
Tarkoituksena on luoda tietojenkäsittelytieteen laitoksen henkilökunnan ja opiskelijoiden käyttöön järjestelmä, jonka avulla voidaan seurata kaikkien harjoitustyökurssien eri työaiheiden suorituksia. Tällä hetkellä eri harjoitustöiden aiheet ovat kaikki omilla kurssikohtaisilla sivuilla. Yksi yhteinen harjoitustyö tietokanta helpottaisi sekä opiskelijoiden että henkilökunnan arkea. Järjestelmän toisena tavoitteena on lisätä harjoitustöiden läpinäkyvyyttä, sen avulla opiskelijat ja henkilökunta saavat helposti käyttöönsä hyödyllistä tilastotietoa harjoitustöiden suoritukseen liittyen. Yliopiston henkilökunta voi lisäksi käyttää järjestelmästä saatavia tilastotietoja opetuksen kehittämiseen.

Järjestelmässä henkilökuntaan kuuluva voi hakea, katsella, lisätä ja muokata sekä töiden aiheita että töiden suorituksia. Lisäksi opettajien on mahdollista poistaa töiden aiheita. Opiskelijoiden oikeudet järjestelmässä ovat rajoitunempia, he voivat ainoastaan hakea ja katsella töiden aiheita ja suorituksia. Opiskelijoiden suoritustietonäkymä on myös suppeampi kuin henkilökuntaan kuuluvan käyttäjän. Harjoitustöitä haetaan järjestelmä erilaisten luokittelutietojen avulla. Järjestelmän ylläpito voi lisätä ja muokata muiden käyttäjien tietoja. Järjestelmän tietojen lisääminen, muokkaaminen ja poisto vaativat sisäänkirjautumisen.

Järjestelmä on tarkoitettu toteuttaa käyttäen PHP ohjelmointikieltä ja PostgreSQL-tietokantaa, ja se tulee sijaitsemaan Helsingin yliopiston tietojenkäsittelytieteen laitoksen users2017 palvelimella. Palvelimen palvelinohjelmistona toimii Apache. Järjestelmässä tulee (luultavasti) olemaan hieman Javascriptillä toteutettua front-endiä, ja järjestelmää on tarkoitettu käyttää webselaimella, joten käyttäjän selaimen täytyy tukea Javascriptiä.

## 2. Yleiskuva järjestelmästä

### 2.1 Käyttötapauskaavio



### 2.2 Käyttäjäryhmät

**Opettaja:** Opettaja on sovelluksen varsinainen kirjautunut käyttäjä, joka voi lisätä, hakea ja poistaa tietoja.

**Jokamies:** Jokamies on sovelluksen kirjautumaton käyttäjä, joka voi vain hakea tietoja.

**Ylläpitäjä:** Ylläpitäjä on sovelluksen kirjautunut käyttäjä, jolla on järjestelmässä oikeudet tehdä kaikkea mitä opettaja voi tehdä, mutta lisäksi ylläpitäjällä on ekstra oikeuksia käyttäjien ja kurssien hallintaan liittyen.

## 2.3 Käyttötapauskuvaukset

### Opettajan käyttötapauskuvaukset

Työaiheiden haku:

Opettaja voi hakea tietokannasta erilaisia rajoituksia käyttäen harjoitustöiden aiheita.

Työaiheiden katselu:

Opettaja voi katsella sovelluksessa olevia työaiheiden näkymiä.

Työaiheiden lisääminen:

Opettaja voi lisätä uusia työaiheita tietokantaan.

Työaiheiden muokkaus:

Opettaja voi muokata tietokannassa jo olevia työaiheita.

Työaiheen poisto:

Opettaja voi poistaa tietokannassa olevan työaiheen.

Suoritustietojen haku:

Opettaja voi hakea tietokannasta erilaisia rajoituksia käyttäen harjoitustöiden suoritustietoja

Suoritustietojen lisääminen:

Opettaja voi lisätä tietokantaan uusia suoritustietoja.

Suoritustietojen muokkaus:

Opettaja voi muokata tietokannasta jo olevia suoritustietoja.

Suoritustietojen poisto:

Opettaja voi poistaa tietokannassa olevia suoritustietoja

Kurssien haku:

Opettaja voi hakea tietokannasta erilaisia rajoituksia käyttäen kurssien tietoja.

Kurssien muokkaus:

Opettaja voi muokata tietokannassa jo olevia kursseja.

Käyttäjien haku:

Opettaja voi hakea tietokannasta erilaisia rajoituksia käyttäen henkilöiden tietoja.

Käyttäjien lisääminen:

Opettaja voi lisätä tietokantaan uusia käyttäjiä.

Muita käyttötapauksia:

Kirjautuminen: opettajan täytyy kirjautua sovellukseen käyttäjätunnuksella ja salasanalla ennen kuin mitään opettajan käyttötapauksia voidaan suorittaa.

Jokamiehen käyttötapaukset

Työaiheiden haku:

Jokamies voi hakea tietokannasta erilaisia rajoituksia käyttäen harjoitustöiden aiheita.

Suoritustietojen haku:

Jokamies voi hakea erilaisia rajoituksia käyttäen harjoitustöiden suoritustietoja.

Kurssien haku:

Jokamies voi hakea tietokannasta erilaisia rajoituksia käyttäen kurssien tietoja.

Käyttäjien haku:

Jokamies voi hakea tietokannasta erilaisia rajoituksia käyttäen henkilöiden tietoja.

Ylläpitäjän käyttötapaukset

Käyttäjien tiedon muokkaus:

Ylläpitäjä voi muokata tietokannassa jo olevien käyttäjien tietoja.

Käyttäjien poisto:

Ylläpitäjä voi poistaa tietokannassa jo olevia käyttäjiä.

Kurssien lisääminen:

Ylläpitäjä voi lisätä tietokantaan uusia kursseja.

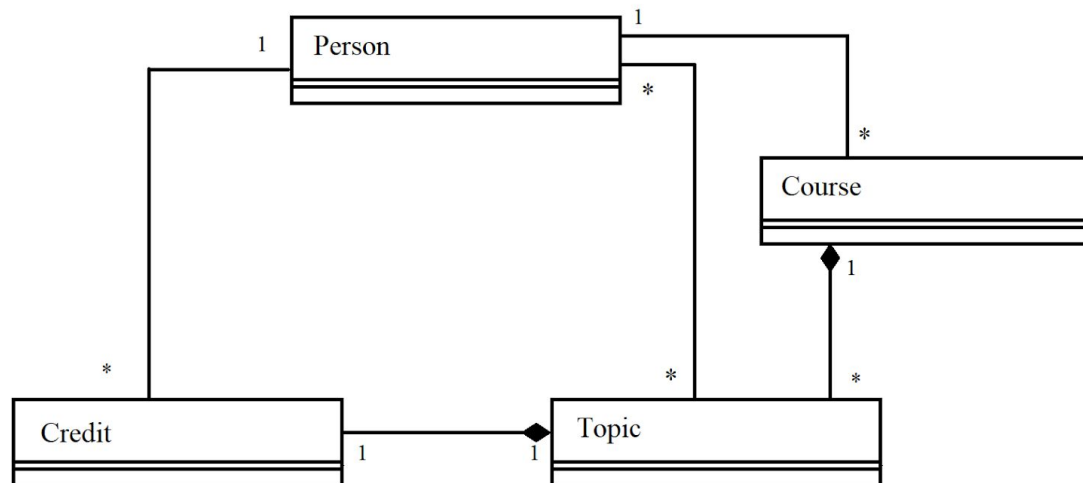
Kurssien poisto:

Ylläpitäjä voi poistaa tietokannassa jo olevia kursseja.

Muita käyttötapauksia:

Kirjautuminen: ylläpitäjän täytyy kirjautua sovellukseen käyttäjätunnuksella ja salasanalla ennen kuin mitään ylläpidon käyttötapauksia voidaan suorittaa.

### 3. Järjestelmän tietosisältö



Tietokohde: Person

Attribuutti	Arvojoukko	Kuvailu
Name	Merkkijono, max 50 merkkiä	Käyttäjän koko nimi.
Username	Merkkijono, max 50 merkkiä	Käyttäjän käyttäjätunnus
Password	Merkkijono, max 50 merkkiä	Käyttäjän salasana.
Admin	Boolean	Tosi jos käyttäjä on ylläpitäjä.

Käyttäjä voi olla, joko ylläpitäjä tai opettaja, uusi käyttäjä on oletusarvoisesti opettaja. Vain ylläpitäjä voi lisätä uusia käyttäjiä tai kursseja, ja muokata käyttäjien tai kurssien tietoja. Käyttäjään voi liittyä yksi tai useampi työaihe ja suoritus. Käyttäjän ja aiheen välillä on liitostaulu. Käyttäjän nimessä, käyttäjätunnuksessa ja salasanassa tyhjämerkkijono ei ole sallittu. Käyttäjätunnuksen on oltava uniikki.

Tietokohde: Course

Attribuutti	Arvojoukko	Kuvailu
Name	Merkkijono, max 50 merkkiä	Kurssin koko nimi.
Incharge	Integer	Kurssin vastuuhenkilön id.

Jokaisella kurssilla on tasan yksi vastuuhenkilö. Kurssin nimi ei saa olla tyhjä.

Tietokohde: Topic

Attribuutti	Arvojoukko	Kuvailu
Name	Merkkijono, max 50 merkkiä	Aiheen koko nimi.
Description	Merkkijono, max 500 merkkiä	Aiheen koko kuvaus.
Course	Integer	Aiheen kurssin id.

Aihe liittyy aina yhteen tai useampaan käyttäjään, jotka vastaavat aiheesta. Aiheen ja käyttäjän välillä on liitostaulu. Aiheen nimessä tyhjämerkkijono ei ole sallittu. Aihe kuuluu aina johonkin tiettyyn kurssiin.

Tietokohde: Credit

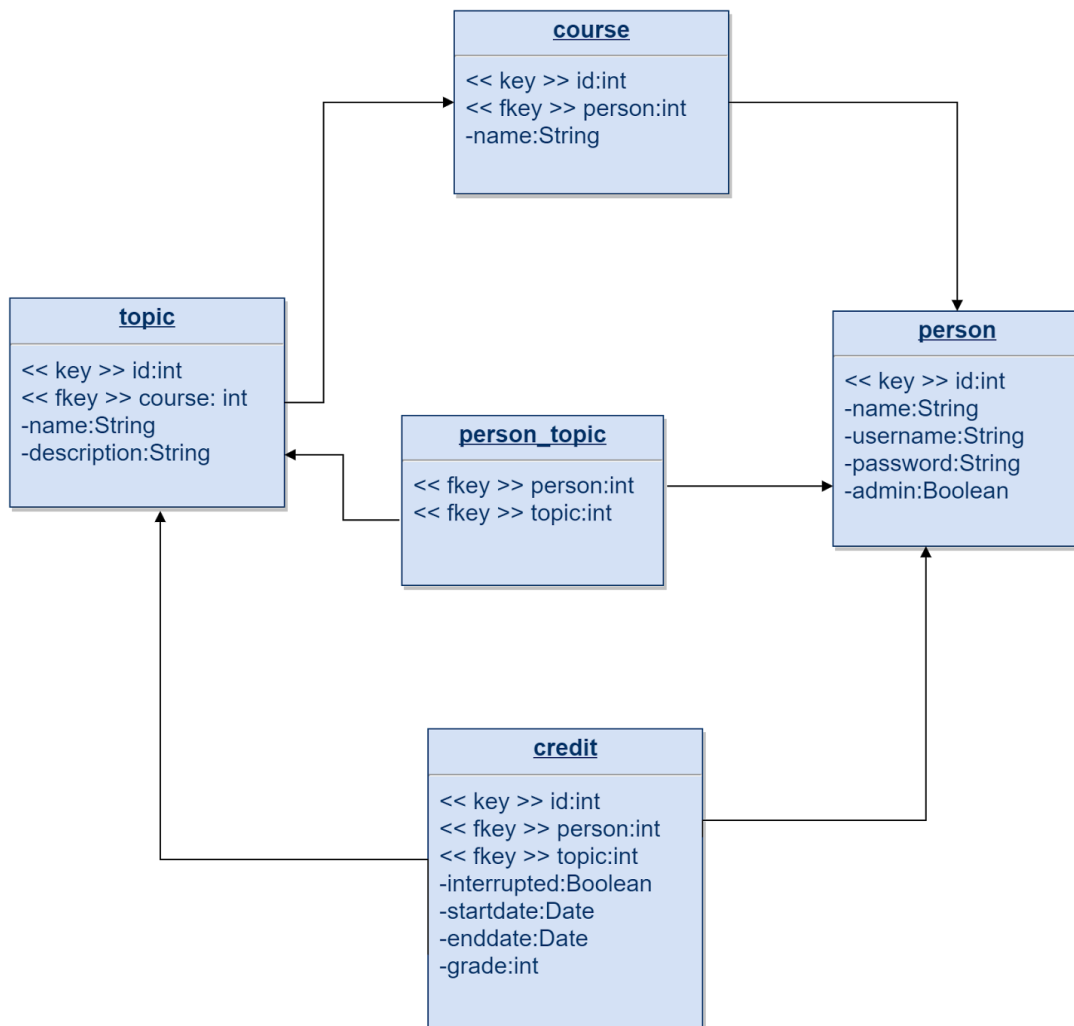
Attribuutti	Arvojoukko	Kuvailu
Givenby	integer	Suorituksen lisäneen käyttäjän id.
Topicid	integer	Suoritukseen liittyvän aiheen id.
Interrupted	boolean	Tosi jos suoritus on keskeytetty.



startdate	date	Päivämäärä jolloin suoritus on aloitettu
enddate	date	Päivämäärä jolloin suoritus on hyväksytty.
Grade	integer	Suorituksesta saatu arvosana.

Suoritukseen liittyy aina yksi aihe ja yksi käyttäjä. Aihe on suorituksessa tehtävän työn aihe. Käyttäjä on suorituksen tietokantaan lisännyt käyttäjä. Suoritus voi olla keskeytetty, oletusarvoisesti suoritus ei ole keskeytetty. Suoritus aloitetaan tiettyinä päivämääränä ja päättyy tiettyinä päivämääränä, jos sitä ei keskeytetä. Päivämäärien erotusta voidaan käyttää suoritusajan laskentaan. Ei keskeytetystä suorituksesta annetaan arvosana.

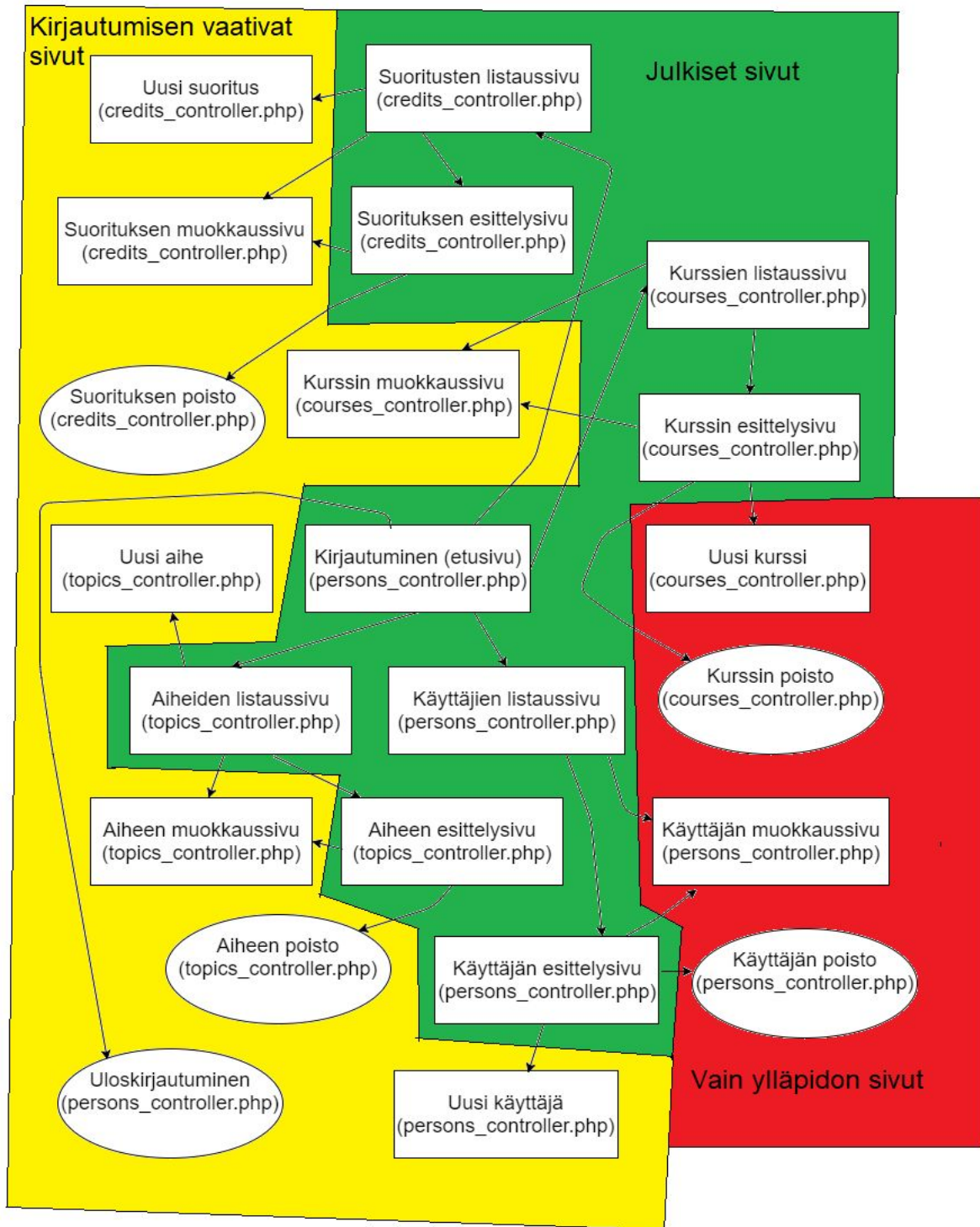
#### 4. Relaatiotietokantakaavio



## 5. Järjestelmän yleisrakenne

Tietokantasovellusta tehdessä on noudatettu MVC-mallia. Kontrollerit, näkymät ja mallit sijaitsevat hakemistoissa `controllers`, `views`- ja `models`. Kaikki sovelluksen kontrollit ja mallit perivät `base_controllerin` ja `base_modelin` toiminnot. Nämä ja muut käytetyt apukirjastot on sijoitettu hakemistoon `lib` ja asetukset ovat hakemistossa `config`. Kansiota `assets/js` löytyvät sovelluksessa käytetyt Javascriptit ja kansiota `views/macros` sovelluksessa käytetyt makrot. Sovelluksen pääsynvalvonta on toteutettu session avulla. Kirjautumattomilla käyttäjillä on pääsy ainoastaan listaus ja esittelynäkyymiin. Kirjautunut käyttäjät voivat lisäksi olla ylläpitäjiä, jolla on tavallisen kirjautuneen käyttäjien oikeuksien lisäksi mahdollisuus lisätä ja poistaa kursseja, sekä muokata ja poistaa käyttäjien tietoja.

## 6. Käyttöliittymä ja järjestelmän komponentit



Sivuilla on navigointipalkki joista päästään kirjautumis / uloskirjautumissivulle ja kaikkiin listaussivuihin. Lisäksi kaavion selkeyttämisen vuoksi siellä ei ole esitelty listaus ja esittelysivun linkkejä. Tämä linkit vievät aina listaus tai esittelysivulla esitetyn tietokohteen attribuuttina olevan tietokohteen esittelysivulle. Esimerkiksi aiheen esittelysivulta pääsee kyseisen aiheen vastuuhenkilön esittelysivulle ja kurssin esittelysivulle.

## 7.1 Asennustiedot ja käyttöohjeet

### 7.1 Asennustiedot

Asenna sovellus kopioimalla sen tiedostot palvelimen internettiin näkyvään hakemistoon (esim. usersin htdocs-hakemisto). Aseta sen jälkeen tietokannan yhteystiedot oikeaksi tiedostoon config/database.php. Tämän lisäksi pitää ajaa käytettävässä tietokannassa create\_tables.sql tiedosto. Halutessa tietokantataulut voidaan poistaa ajamalla drop\_tables.sql tiedosto. Testidataa tietokantaan voi lisätä ajamalla (testidataa voi kyseiseen tiedostoon myös itse lisätä) add\_test\_data.sql tiedosto.

### 7.2 Käynnistys- / käyttöohje

Sovellusta voidaan demokäyttää osoitteessa: <http://andra.users.cs.helsinki.fi/tsoha/>

Tietoja pääsee selailemaan ilman kirjautumista, mutta vain kirjautuneet käyttäjät voivat lisätä, muokata tai poistaa tietoja. Kirjautuneet käyttäjät jaetaan kahteen ryhmään, tavallisiin käyttäjiin ja ylläpitäjään. Ylläpitäjällä on kaikki oikeudet mitkä tavallisella käyttäjällä, mutta tällä hetkellä vain ylläpitäjät voivat lisätä ja poistaa kursseja ja lisätä, muokata ja poistaa käyttäjiä.

Sovellusta voidaan käyttää seuraavilla käyttäjätunnus ja salasana pareilla. Ohja -tunnuksella on ylläpitäjän oikeudet, Ope3 ei.

Tunnus	Salasana
Ohja	aja
Ope3	Ope3

## 8. Testaus, tunnetut bugit ja puutteet, jatkokehitysideat & muu dokumentaatio

Sovellusta on testattu ainoastaan käsin pääasiassa itseni toimesta. Tällä hetkellä sovelluksessa ei ole tiedossa olevia bugeja, mutta allekirjoittanut ei voi varmasti sanoa etteikö niitä olisi.

Tämä sovellus on alusta alkaen suunniteltu minimaalisesti ja modulaarisesti (niin paljon kun MVC arkkitehtuurimallissa voi) niin, että sinne olisi mahdollista suhteellisen helposti lisätä toiminnallisuutta. Mahdolliset jatkokehitysideat olisivat 1) tietokantaan tauluihin lisää attribuutteja, jonka avulla saataisiin jonkinlaista loppukäyttäjää innostavaa dataa (esimerkiksi kurssi-taulu on tarkoituksella jätetty hyvin tyhjäksi). 2) sovellukseen voi myös lisätä avain uusia tauluja käyttäjän mielenkiinnon mukaan tai sen voisi integroida johonkin muuhun yliopisto-opiskelua helpottavaa järjestelmään. Näissä tapauksissa on syytä muistaa olla tarkka ettei pääse syntymään single point of failure / pullonkauloja.

### Omat kokemukset

-Kurssi oli minulle teknisiltä osa-aloilta melko helppo, vaikein osio oli uuden kielen (php opettelu) ja tämän dokumentaation tekeminen. Erityisesti tuskaa tuotti dokumentaatiossa olevien piirrosten tuottaminen, syynä lienee se, että olen graafisesti lahjaton ja minulla on vain vähän kokemusta piirtämiseen käytettävistä ohjelmistoista.

-Lisää olisin toivonut testaamista, ei yksikkötestausta vaan integraatio ja hyväksymätestausta. Kuitenkin työelämässä ohjelmiston testaus on erittäin tärkeä osa ja sitä opetellaan varsin niukasti laitoksen kursseilla.

-Demo olisi ollut kivempi hieman isommissa ryhmissä, minua olisi kiinnostanut nähdä mitä jännää muut opiskelijat ovat saaneet aikaan.

-Kokonaisuudessaan yksi laitoksen parhaimpia kursseja, harmittaa että itse kävin niin myöhäisessä vaiheessa opiskelua.